REPORT DOCUMENTATION PAGE

Form Approved
OMB NO. 0704-0188

1. AGENCY USE ONLY (Leave Blank)

2. REPORT DATE 12/14/01

3. REPORT TYPE AND DATES COVERED
Final Progress Report 12/95 – 09/01
11Dec95-30Sep01

4. TITLE AND SUBTITLE
Final Technical Report
Applicable and Robust Geometric Computing

5. FUNDING NUMBERS
DAAH04-96-1-0013

6. AUTHOR(S)
Franco P. Preparata (Principal Investigator),
Pankaj K. Agarwal, Michael T. Goodrich, S. Rao Kosaraju,
Roberto Tamassia, Jeffrey S. Vitter

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Brown University, Providence, RI (main contractor)
Duke University, Durham, NC (subcontractor)
Johns Hopkins University, Baltimore, MD (subcontractor)

8. PERFORMING ORGANIZATION
REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U. S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSORING / MONITORING
AGENCY REPORT NUMBER

34990.65-MA-MUR

11. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.

12 a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12 b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

This research project is aimed at aimed at facilitating an effective technology transfer from computational geometry to the various applied fields to which it is relevant. Our technical contributions include algorithmic foundations, practical methodologies, emerging technologies, and applications.

14. SUBJECT TERMS
Computational geometry, algorithms, software libraries

15. NUMBER OF PAGES
39

16. PRICE CODE

17. SECURITY CLASSIFICATION OR REPORT	18. SECURITY CLASSIFICATION ON THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Final Technical Report

ARO Grant DAAH04–96–1–0013
Applicable and Robust Geometric Computing

Pankaj K. Agarwal[2] *Michael T. Goodrich*[3] *S. Rao Kosaraju*[3]

Franco P. Preparata[1*] *Roberto Tamassia*[1] *Jeffrey S. Vitter*[2]

(December 14, 2001)

1. Center for Geometric Computing, Department of Computer Science,
Brown University,
115 Waterman Street, Providence, RI 02912–1910,
`http://www.cs.brown.edu/cgc-brown.html`.
2. Center for Geometric Computing, Department of Computer Science,
Duke University,
Levine Science Research Center, Durham, NC 27708–0129,
`http://www.cs.duke.edu/CGC/`.
3. Center for Geometric Computing, Department of Computer Science,
The Johns Hopkins University,
3400 N. Charles Street, Baltimore, MD 21218,
`http://www.cs.jhu.edu/labs/cgc/`.

*__Principal Investigator.__ Tel: (401) 863–7649. Fax: (401) 863–7657. E-mail: `franco@cs.brown.edu`.

1 Scientific Personnel

1.1 Brown University

Faculty:

- Franco P. Preparata (professor)
- Roberto Tamassia (professor)

Postdoctoral Research Associates

- Ulrik Brandes
- Ashim Garg
- Giuseppe Liotta

Research Associates and Consultants

- David Emory
- Maurizio Pizzonia
- Luca Vismara

Graduate Students:

- James E. Baker
- Stina Bridgeman
- Vasiliki Chatzi
- Galina Shubina
- Nikos Triandopoulos

Undergraduate Students:

- Ryan Baker
- Michael Boilen
- Devin Borland
- David Emory
- Natasha Gelfand
- Mark Handy
- Benoit Hudson
- Lucy Perry
- Keith Schmidt
- Andrew Schwerin

Visiting Scientists:

- Jean-Daniel Boissonnat (I.N.R.I.A., Sophia-Antipolis, France)
- Ulrik Brandes (University of Konstanz, Germany)
- Robert Cohen (AlgoMagic Technologies, Inc., MA)
- Isabel Cruz (University of Illinois, Chicago)
- Olivier Devillers (I.N.R.I.A., Sophia-Antipolis, France)
- Giuseppe Di Battista (University Rome, Italy)
- Peter Eades (University of Newcastle, Australia)
- Lutz Kettner (University of North Carolina)
- Michael T. Goodrich (University of California, Irvine)
- Giuseppe Liotta (University of Perugia)
- Kokichi Sugihara (University of Tokio)
- Ioannis G. Tollis (University of Texas at Dallas)
- Paola Vocca (University of Rome "Tor Vergata," Italy)

1.2 Duke University

Faculty

- Pankaj K. Agarwal, professor;
- Lars Arge, assistant professor;
- Jeffrey S. Vitter, professor;

Postdoctoral Research Associates

- Lars Arge
- Julien Basch
- David Hutchinson
- Sariel Har-Peled

Graduate Students

- Rakesh Barve
- Anne Collins
- Pavan Desikan
- Sathish Govindarajan
- Lipyeow Lim
- Stacy Luoma
- T. M. Murali,
- Nabil Hassan Mustafa
- Paul Natsev
- Magda Procopiuc
- Octavian Procopiuc
- Laura Toma
- Kasturi Varadarajan
- Darren Vengroff
- Rajiv Wickremesinghe
- Min Wang
- Yusu Wang

Undergraduate Students

- Mark Bauman
- Alexander Chrisman
- Meetesh Karia

- Alexander Karwait
- Antoine Kremer
- Jay Steele
- Michael Strauss
- John Tran

Visiting Scientists:

- Mark de Berg
- Jeff Erickson
- Leonidas Guibas
- Dan Halperin
- Christian S. Jensen
- Mark Overmars
- Peter Sanders
- Micha Sharir
- Peter Varman

1.3 Johns Hopkins University

Faculty:

- S. Rao Kosaraju, Professor
- Michael T. Goodrich, Professor
- Subodh Kumar, Assistant Professor

Research Associates and Consultants

- Robert Cohen

Postdoctoral Research Associates

- Gill Barequet
- Timothy Chan
- Pawel Gajer

Graduate Students:

- Ryan Borgstrom
- Michael Capalbo
- Amitabh Chaudhary
- Christian Duncan
- Wenjing Huang
- Stephen Kobourov
- John Kloss
- Jeremy Mullendore
- Mihai Pop
- Kumar Ramaiyer
- Christopher Riley
- Jing Zhu

Undergraduate Students:

- Edward Bross
- Michael Cohen
- Matthew Harris
- Samuel Lang
- Joel Sandin
- Russell Schwagger

- Aaron Stibel
- David Tucker

Visiting Scientists:

- Matthew Dickerson (Middlebury College)
- Adam Buchsbaum (AT&T Research Labs)

2 Publications

The following publications all acknowledge ARO Grant DAAH04–96–1–0013. For brevity, we do not list papers published in conference proceedings for which a journal version is also available.

[1] P. K. Agarwal. Range searching. In J. E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, chapter 31, pages 575–598. CRC Press LLC, Boca Raton, FL, 1997.

[2] P. K. Agarwal, N. Amenta, B. Aronov, and M. Sharir. Largest placements and motion planning of a convex polygon. In J.-P. Laumond and M. Overmars, editors, *Algorithms for Robotic Motion and Manipulation*, pages 143–154, Wellesley, Massachusetts, 1997. A.K. Peters. Proc. 1996 Workshop on the Algorithmic Foundations of Robotics.

[3] P. K. Agarwal, N. Amenta, and M. Sharir. Placement of one convex polygon inside another. *Discrete Comput. Geom.*, 19:95–104, 1998.

[4] P. K. Agarwal, L. Arge, G. Brodal, and J. S. Vitter. I/o-efficient dynamic point location in monotone subdivisions. In *Proc. 10th ACM-SIAM Sympos. Discrete Algorithms*, pages 11–20, 1999.

[5] P. K. Agarwal, L. Arge, and J. Erickson. Indexing moving points. In *Proc. Annu. ACM Sympos. Principles Database Syst.*, 2000. 175–186.

[6] P. K. Agarwal, L. Arge, J. Erickson, P. G. Franciosa, and J. S. Vitter. Efficient searching with linear constraints. *Journal of Computer and System Sciences*, pages 194–216, October 2000.

[7] P. K. Agarwal, L. Arge, T. Murali, K. Varadarajan, and J. Vitter. I/O-efficient algorithms for contour line extraction and planar graph blocking. In *Proc. ACM-SIAM Symp. on Discrete Algorithms*, pages 117–126, 1998.

[8] P. K. Agarwal, L. Arge, O. Procopiuc, and J. S. Vitter. A framework for index bulk loading and dynamization. In *Proc. 28th Intl. Conference on Automata and Programming Langs.*, 2001.

[9] P. K. Agarwal, L. Arge, and J. Vahrenhold. Time responsive indexing schemes for moving points. In *Proc. 7th Workshop on Data Structures*, Lecture Notes Comput. Sci. Springer-Verlag, 2001.

[10] P. K. Agarwal, B. Aronov, T. M. Chan, and M. Sharir. On levels in arrangements of lines, segments, planes, and triangles. *Discrete Comput. Geom.*, 19:315–331, 1998.

[11] P. K. Agarwal, B. Aronov, S. Har-Peled, and M. Sharir. Approximation and exact algorithms for minimum-width annuli and shells. In *Proc. 15th Annu. ACM Sympos. Comput. Geom.*, pages 380–389, 1999.

[12] P. K. Agarwal, B. Aronov, J. O'Rourke, and C. A. Schevon. Star unfolding of a polytope with applications. *SIAM J. Comput.*, 26:1689–1713, 1997.

[13] P. K. Agarwal, B. Aronov, J. Pach, R. Pollack, and M. Sharir. Quasi-planar graphs have a linear number of edges. *Combinatorica*, 17:1–9, 1997.

[14] P. K. Agarwal, B. Aronov, and M. Sharir. Line traversals of balls and smallest enclosing cylinders in three dimensions. In *Proc. 8th ACM-SIAM Sympos. Discrete Algorithms*, pages 483–492, 1997.

[15] P. K. Agarwal, B. Aronov, and M. Sharir. On levels in arrangements of lines, segments, planes and triangles. In *Proc. ACM Symp. on Computational Geometry*, pages 30–38, 1997.

[16] P. K. Agarwal, B. Aronov, and M. Sharir. Motion planning for a convex polygon in a polygonal environment. *Discrete Comput. Geom.*, 22:201–221, 1999.

[17] P. K. Agarwal, B. Aronov, and M. Sharir. Exact and approximation algorithms for minimum-width cylindrical shells. In *Proc. 11th ACM-SIAM Sympos. Discrete Algorithms*, pages 510–517, 2000.

[18] P. K. Agarwal, B. Aronov, and M. Sharir. On the complexity of many faces in arrangements of circles. In *Proc. 41st Annual IEEE Sympos. on Foundations of Computer Science*, 2001.

[19] P. K. Agarwal, J. Basch, L. Guibas, and J. Hershberger. Lower bounds for kinetic planar subdivisions. *Discrete Comput Geom.*, 24:721–733, 2000.

[20] P. K. Agarwal, J. Basch, L. J. Guibas, J. Hershberger, and L. Zhang. Deformable free space tiling for kinetic collision detection. In *Proc. 4th Workshop Algorithmic Found. Robot.*, 2000. To appear.

[21] P. K. Agarwal, T. Biedl, S. Lazard, S. Robbins, S. Suri, and S. Whitesides. Curvature-constrained shortest paths in a convex polygon. In *Proc. 14th Annu. ACM Sympos. Comput. Geom.*, page to appear, 1998.

[22] P. K. Agarwal, M. de Berg, J. Gudmundsson, M. Hammar, and H. J. Haverkort. Box-trees and r-trees with near-optimal query time. In *Proc. 17th Annu. Symposium on Computational Geometry*, 2001.

[23] P. K. Agarwal, M. de Berg, D. Halperin, and M. Sharir. Efficient generation of k-directional assembly sequences. In *Proc. 7th ACM-SIAM Sympos. Discrete Algorithms*, pages 122–131, 1996.

[24] P. K. Agarwal, M. de Berg, S. Har-Peled, M. Overmars, M. Sharir, and J. Vahrenhold. Reporting all intersecting pairs of polytopes in two and three dimensions. In *Proc. 7th Workshop on Data Structures*, Lecture Notes Comput. Sci. Springer-Verlag, 2001.

[25] P. K. Agarwal and P. K. Desikan. An efficient algorithm for terrain simplification. In *Proc. 8th ACM-SIAM Sympos. Discrete Algorithms*, pages 139–147, 1997.

[26] P. K. Agarwal and P. K. Desikan. Approximation algorithms for layered manufacturing. In *Proc. 11th ACM-SIAM Sympos. Discrete Algorithms*, pages 528–537, 2000.

[27] P. K. Agarwal, D. Eppstein, L. J. Guibas, and M. Henzinger. Parametric and kinetic minimum spanning trees. In *Proc. 39th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 596–605, 1998.

[28] P. K. Agarwal and J. Erickson. Geometric range searching and its relatives. In B. Chazelle, J. E. Goodman, and R. Pollack, editors, *Advances in Discrete and Computational Geometry*, volume 223 of *Contemporary Mathematics*, pages 1–56. American Mathematical Society, Providence, RI, 1999.

[29] P. K. Agarwal, J. Erickson, and L. J. Guibas. Kinetic BSPs for intersecting segments and disjoint triangles. In *Proc. 9th ACM-SIAM Sympos. Discrete Algorithms*, pages 107–116, 1998.

[30] P. K. Agarwal, E. Flato, and D. Halperin. Polygon decomposition for efficient construction of Minkowski sums. In *Proc. 8th Annu. European Sympos. Algorithms*, volume 1879 of *Lecture Notes Comput. Sci.*, pages 20–31. Springer-Verlag, 2000.

[31] P. K. Agarwal, E. F. Grove, T. M. Murali, and J. S. Vitter. Binary space partitions for fat rectangles. *SIAM J. Comput.*, 29:1422–1448, 2000.

[32] P. K. Agarwal, L. J. Guibas, S. Har-Peled, A. Rabinovitch, and M. Sharir. Computing the penetration depth of two convex polytopes in 3d. In *Proc. 7th Scandanavian Workshop on Algorithmic Theory*, page ??, 2000.

[33] P. K. Agarwal, L. J. Guibas, J. Hershberger, and E. Veach. Maintaining the extent of a moving point set. In *Proc. 5th Workshop Algorithms Data Struct.*, volume 1272 of *Lecture Notes Comput. Sci.*, pages 31–44. Springer-Verlag, 1997.

[34] P. K. Agarwal, L. J. Guibas, T. M. Murali, and J. S. Vitter. Cylindrical static and kinetic binary space partitions. *Computational Geometry*, 16(2):103–127, 2000.

[35] P. K. Agarwal and S. Har-Peled. Maintaining approximate extent measures of moving points. In *Proc. 12th ACM-SIAM Sympos. Discrete Algorithms*, 2001.

[36] P. K. Agarwal, S. Har-Peled, and M. Karia. Computing approximate shortest paths on convex polytopes. In *Proc. 16th Annu. ACM Sympos. Comput. Geom.*, pages 270–279, 2000.

[37] P. K. Agarwal, S. Har-Peled, M. Sharir, and K. R. Varadarajan. Approximate shortest paths on a convex polytope in three dimensions. *J. ACM*, 44:567–584, 1997.

[38] P. K. Agarwal, S. Har-Peled, and Y. Wang. Occlusion culling for fast walkthrough in urban areas. In *Eurographics*, 2001.

[39] P. K. Agarwal, L. Knipping, M. van Kreveld, T. Strijk, and A. Wolf. A simple and efficient algorithm for high-quality line labeling. In *Abstracts 15th European Workshop Comput. Geom.*, pages 93–96. INRIA Sophia-Antipolis, 1999.

[40] P. K. Agarwal, J.-C. Latombe, R. Motwani, and P. Raghavan. Nonholonomic path planning for pushing a disk among obstacles. In *Intl. Conf. Robotics and Auto.*, 1997.

[41] P. K. Agarwal, T. Murali, and J. Vitter. Practical techniques for constructing binary space partitions for orthogonal rectangles. In *Proc. 13th Annu. ACM Sympos. Comput. Geom.*, pages 382–384, 1997.

[42] P. K. Agarwal, T. Murali, and J. Vitter. A new algorithm for constructing binary space partitions for orthogonal rectangles. In *Proc. European Sympos. Algortihms*, 1998.

[43] P. K. Agarwal and C. M. Procopiuc. Exact and approximation algorithms for clustering. In *Proc. 9th ACM-SIAM Sympos. Discrete Algorithms*, pages 658–667, 1998.

[44] P. K. Agarwal and C. M. Procopiuc. Approximation algorithms for projective clustering. In *Proc. 11th ACM-SIAM Sympos. Discrete Algorithms*, pages 538–547, 2000.

[45] P. K. Agarwal and S. Sen. Randomized algorithms for geometric optimization. In *Handbook on Randomized Computation.* Kluwer Academic Publishers, 2001.

[46] P. K. Agarwal and M. Sharir. Efficient algorithms for geometric optimization. *ACM Comput. Surv.*, 30:412–458, 1998.

[47] P. K. Agarwal and M. Sharir. Motion planning of a ball amid polyhedral obstacles in three dimensions. In *Proc. 10th ACM-SIAM Sympos. Discrete Algorithms*, pages 21–30, 1999.

[48] P. K. Agarwal and M. Sharir. Arrangements and their applications. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 49–119. Elsevier Science Publishers B.V. North-Holland, Amsterdam, 2000.

[49] P. K. Agarwal and M. Sharir. Davenport-Schinzel sequences and their geometric applications. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 1–47. Elsevier Science Publishers B.V. North-Holland, Amsterdam, 2000.

[50] P. K. Agarwal and M. Sharir. Pipes, cigars, and kreplach: The union of Minkowski sums in three dimensions. *Discrete Comput. Geom.*, 24(4):645–685, 2000.

[51] P. K. Agarwal and M. Sharir. On the number of congruent simplices in a point set. In *Proc. 17th Sympos. Comput. Geom.*, 2001.

[52] P. K. Agarwal and M. Sharir. Pseudoline arrangements: Duality, algorithms, and applications. In *Proc. 13th ACM-SIAM Sympos. Discrete Algorithms*, 2002.

[53] P. K. Agarwal, M. Sharir, and E. Welzl. The discrete 2-center problem. In *Proc. ACM Symp. on Computational Geometry*, pages 147–155, 1997.

[54] P. K. Agarwal and S. Suri. Surface approximation and geometric partitions. *SIAM J. Comput.*, 27:1016–1035, 1998.

[55] P. K. Agarwal, M. van Kreveld, and S. Suri. Label placement by maximum independent set in rectangles. In *Proc. 9th Canad. Conf. Comput. Geom.*, pages 233–238, 1997.

[56] P. K. Agarwal and K. R. Varadarajan. Efficient algorithms for polygon simplification. *Discrete Comput. Geom.*, 23:273–291, 2000.

[57] C. C. Aggarwal, C. M. Procopiuc, J. L. Wolf, P. S. Yu, and J. S. Park. Fast algorithms for projected clustering. In *Proc. ACM SIGMOD International Conference*, pages 61–72, 1999.

[58] D. Aliaga, F. Brooks, J. Cohen, S. Kumar, M. Lin, D. Manocha, A. Wilson, and H. Zhang. Interactive walkthroughs of large geometric datasets, 1999.

[59] N. M. Amato, M. T. Goodrich, and E. A. Ramos. Computing the arrangement of curve segments: divide-and-conquer algorithms via sampling. In *Proc. 11th ACM-SIAM Sympos. Discrete Algorithms*, pages 705–706, 2000.

[60] N. M. Amato, M. T. Goodrich, and E. A. Ramos. Linear-time triangulation of a simple polygon made easier via randomization. In *Proc. 16th Annu. ACM Sympos. Comput. Geom.*, pages 201–212, 2000.

[61] L. Arge, J. S. Chase, L. Toma, J. S. Vitter, and R. Wickremesinghe. Flow computation on massive grids. In *Proceedings of the ACM International Symposium on Advances in Geographic Information Systems*, volume 9, Atlanta, November 2001.

[62] L. Arge, J. S. Chase, L. Toma, J. S. Vitter, R. Wickremesinghe, P. Halpin, and D. Urban. Digital terrain analysis for massive grids. In *Proceedings of the Annual Symposium of the U.S. Chapter of International Association of Landscape Ecology*, volume 16 of *Pattern, Process, Scale, & Hierarchy: Interactions in Human-Dominated and Natural Landscapes*, Tempe, AZ, April 2001.

[63] L. Arge, K. H. Hinrichs, J. Vahrenhold, and J. S. Vitter. Efficient bulk operations on dynamic R-trees. In *Workshop on Algorithm Engineering and Experimentation*, volume 1619 of *Lecture Notes in Computer Science*, pages 328–348, Baltimore, January 1999. Springer-Verlag.

[64] L. Arge, O. Procopiuc, S. Ramaswamy, T. Suel, J. Vahrenhold, and J. S. Vitter. A unified approach for indexed and non-indexed spatial joins. In *Proceedings of the 7th International Conference on Extending Database Technology*, Konstanz, Germany, March 2000.

[65] L. Arge, O. Procopiuc, S. Ramaswamy, T. Suel, and J. S. Vitter. Theory and practice of I/O-efficient algorithms for multidimensional batched searching problems. In *Proc. ACM-SIAM Symp. on Discrete Algorithms*, pages 685–694, 1998.

[66] L. Arge, V. Samoladas, and J. S. Vitter. Two-dimensional indexability and optimal range search indexing. In *Proceedings of the ACM Conference Principles of Database Systems*, volume 18, pages 346–357, Philadelphia, PA, May–June 1999.

[67] L. Arge, L. Toma, and J. S. Vitter. I/O-efficient algorithms for problems on grid-based terrains. In *Workshop on Algorithm Engineering and Experimentation*, Lecture Notes in Computer Science, San Francisco, January 2000. Springer-Verlag.

[68] L. Arge, D. E. Vengroff, and J. S. Vitter. External-memory algorithms for processing line segments in geographic information systems. *Algorithmica*, to appear.

[69] L. Arge and J. S. Vitter. Optimal dynamic interval management in external memory. In *Proceedings of the IEEE Symposium on Foundations of Computer Science*, volume 37, pages 560–569, Burlington, VT, October 1996.

[70] J. E. Baker, I. F. Cruz, G. Liotta, and R. Tamassia. A new model for algorithm animation over the WWW. *ACM Comput. Surv.*, 27(4):568–572, 1995.

[71] J. E. Baker, I. F. Cruz, G. Liotta, and R. Tamassia. Algorithm animation over the World Wide Web. In *Proc. Int. Workshop on Advanced Visual Interfaces*, pages 203–212, 1996.

[72] J. E. Baker, I. F. Cruz, G. Liotta, and R. Tamassia. The Mocha algorithm animation system. In *Proc. Int. Workshop on Advanced Visual Interfaces*, pages 248–250, 1996.

[73] J. E. Baker, I. F. Cruz, G. Liotta, and R. Tamassia. Visualizing geometric algorithms over the Web. *Comput. Geom. Theor. Appl.*, 12:125–152, 1999.

[74] G. Barequet. Dcel: A polyhedral database and programming environment. *Int. J. of Computational Geometry and Applications.* to appear.

[75] G. Barequet. Using geometric hashing to repair cad objects. *IEEE Computational Science & Engineering (CS&E)*, 4(4):22–28, 1997.

[76] G. Barequet. A lower bound for Heilbronn's triangle problem in d dimensions. In *Proc. 10th Ann. ACM-SIAM Symp. on Discrete Algorithm*, pages 76–81, Baltimore, MD, 1999.

[77] G. Barequet, P. Bose, and M. T. Dickerson. Optimizing constrained offset and scaled polygonal annuli. In *Proc. 6th Workshop on Algorithms and Data Structures (WADS)*, volume 1663 of *Lecture Notes in Computer Science*, pages 62–73. Springer-Verlag, 1999.

[78] G. Barequet, S. S. Bridgeman, C. A. Duncan, M. T. Goodrich, and R. Tamassia. Geometric computing over the internet. *IEEE Internet Computing*, 3(2):21–29, 1999.

[79] G. Barequet, A. Briggs, M. Dickerson, C. Dima, and M. T. Goodrich. Animating the polygon-offset distance function. In *Proc. 13th ACM Symp. on Computational Geometry*, pages 479–480, 1997.

[80] G. Barequet, A. Briggs, M. Dickerson, and M. T. Goodrich. Offset-polygon placement problems. *Computational Geometry: Theory and Applications*, 11(3–4):125–141, 1999.

[81] G. Barequet, M. Dickerson, , and D. Eppstein. On triangulating three-dimensional polygons. *Computational Geometry: Theory and Applications.* to appear.

[82] G. Barequet, M. Dickerson, and R. Drysdale. 2-point site Voronoi diagrams. In *Proc. 6th Workshop on Algorithms and Data Structures (WADS)*, volume 1663 of *Lecture Notes Comput. Sci.*, pages 219–230. Springer-Verlag, 1999.

[83] G. Barequet, M. Dickerson, and M. T. Goodrich. Voronoi diagrams for polygon-offset distance functions. In *Workshop on Algorithms and Data Structures (WADS)*, pages 200–209, 1997.

[84] G. Barequet, M. Dickerson, and P. Pau. Translating a convex polygon to contain a maximum number of points. *Computational Geometry: Theory and Applications*, 8(4):167–179, 1997.

[85] G. Barequet, C. Duncan, M. Goodrich, S. Kumar, and M. Pop. Efficient perspective-accurate silhouette computation. In *ACM symposium on Computational Geometry (Video review)*, Miami, Florida, 1999.

[86] G. Barequet, C. Duncan, M. T. Goodrich, W. Huang, S. Kumar, and M. Pop. Efficient perspective-accurate silhouette computation. In *Fourth CGC workshop on Computational Geometry*, Baltimore, October 1999.

[87] G. Barequet and S. Har-Peled. Efficiently approximating the minimum-volume bounding box of a point set in three dimensions. In *Proc. 10th Ann. ACM-SIAM Symp. on Discrete Algorithms*, pages 82–91, 1999.

[88] G. Barequet and S. Har-Peled. Polygon-containment and translational min-hausdorff-distance between segment sets are 3SUM-hard. In *Proc. 10th Ann. ACM-SIAM Symp. on Discrete Algorithms*, pages S862–863, 1999.

[89] G. Barequet and Y. Kaplan. A data front-end for layered manufacturing. *Computer-Aided Design.* to appear.

[90] G. Barequet and S. Kumar. Repairing cad models. In *Proc. IEEE Visualization*, pages 363–370, 1997.

[91] G. Barequet and M. Sharir. Partial surface and volume matching in three dimensions. *Trans. on Pattern Analysis and Machine Intelligence*, 19(9):929–948, 1997.

[92] R. D. Barve, E. F. Grove, and J. S. Vitter. A competitive application-controlled paging algorithm for a shared cache. *SIAM Journal on Computing*, 20(4):1290–1303, August 2000.

[93] R. D. Barve, M. Kallahalla, P. J. Varman, and J. S. Vitter. Competitive analysis of buffer management algorithms. *Journal of Algorithms*, 36, August 2000.

[94] R. D. Barve and J. S. Vitter. A theoretical framework for memory-adaptive algorithms. In *Proceedings of the IEEE Symposium on Foundations of Computer Science*, volume 40, pages 273–284, New York, October 1999.

[95] J. Bazik, R. Tamassia, S. P. Reiss, and A. van Dam. Software visualization in teaching at Brown University. In J. Stasko, J. Domingue, M. H. Brown, and B. A. Price, editors, *Software Visualization: Programming as a Multimedia Experience*, pages 383–398. MIT Press, 1998.

[96] P. Bertolazzi, G. Di Battista, C. Mannino, and R. Tamassia. Optimal upward planarity testing of single-source digraphs. *SIAM J. Comput.*, 27(1):132–169, 1998.

[97] J.-D. Boissonnat and F. P. Preparata. Robust plane sweep for intersecting segments. *SIAM J. Comput.*, 29(5):1401–1421, 2000.

[98] U. Brandes, G. Shubina, and R. Tamassia. Improving angular resolution in visualizations of geographic networks. In *Proc. Joint Eurographics — IEEE TCVG Symposium on Visualization (VisSym '00)*, 2000.

[99] U. Brandes, G. Shubina, R. Tamassia, and D. Wagner. Fast layout methods for timetable graphs. In *Graph Drawing (Proc. GD 2000)*, Lecture Notes in Computer Science. Springer-Verlag, 2000.

[100] S. Bridgeman, A. Garg, and R. Tamassia. A graph drawing and translation service on the WWW. *Internat. J. Comput. Geom. Appl.*, 9(4–5):419–446, 1999.

[101] S. Bridgeman, M. T. Goodrich, S. G. Kobourov, and R. Tamassia. PILOT: An interactive tool for learning and grading. In *Proc. ACM Tech. Symp. on Computer Science Education (SIGCSE)*, 2000.

[102] S. Bridgeman and R. Tamassia. Difference metrics for interactive orthogonal drawing. *J. Graph Alg. Appl.*, 4(3):47–74, 2000.

[103] S. Bridgeman and R. Tamassia. A user study in similarity measures for graph drawing. In *Graph Drawing (Proc. GD 2000)*, Lecture Notes in Computer Science. Springer-Verlag, 2000.

[104] S. S. Bridgeman, G. Di Battista, W. Didimo, G. Liotta, R. Tamassia, and L. Vismara. Turn-regularity and optimal area drawings of orthogonal representations. *Comput. Geom. Theory Appl.*, 16(1):53–93, 2000.

[105] S. S. Bridgeman, J. Fanto, A. Garg, R. Tamassia, and L. Vismara. InteractiveGiotto: An algorithm for interactive orthogonal graph drawing. In G. Di Battista, editor, *Graph Drawing (Proc. GD '97)*, volume 1353 of *Lecture Notes Comput. Sci.*, pages 303–308. Springer-Verlag, 1997.

[106] A. L. Buchsbaum, M. T. Goodrich, and J. R. Westbrook. Range searching over tree cross products. In *Proc. 8th European Symp. on Algorithms*, volume 1879 of *Lecture Notes Comput. Sci.*, pages 120–131. Springer-Verlag, 2000.

[107] M. R. Capalbo. A small universal graph for bounded-degree planar graphs. In *Proc. 10th Ann. ACM-SIAM Symp. on Discrete Algorithms*, pages 156–160, 1999.

[108] M. R. Capalbo and S. R. Kosaraju. Small universal graphs. In *Proc. 21st ACM Symp. on Theory of Computing (STOC)*, pages 741–749, 1999.

[109] T. M. Chan, M. T. Goodrich, S. R. Kosaraju, and R. Tamassia. Optimizing area and aspect ratio in straight-line orthogonal tree drawings. In S. North, editor, *Graph Drawing (Proc. GD '96)*, volume 1190 of *Lecture Notes Comput. Sci.*, pages 63–75. Springer-Verlag, 1997.

[110] C. C. Cheng, C. A. Duncan, M. T. Goodrich, and S. G. Kobourov. Drawing planar graphs with circular arcs. In *Proc. Graph Drawing*, volume 1731 of *Lecture Notes in Computer Science*, pages 117–126, 1999.

[111] L. Chew, M. T. Goodrich, D. H. K. Kedem, J. Kleinberg, and D. Kravets. Geometric pattern matching under euclidean motion. *Computational Geometry: Theory and Applications*, 7:113–124, 1997.

[112] Y.-J. Chiang and R. Tamassia. Optimal shortest path and minimum-link path queries between two convex polygons inside a simple polygonal obstacle. *Internat. J. Comput. Geom. Appl.*, 7:85–121, 1997.

[113] M. Chrobak, M. T. Goodrich, and R. Tamassia. Convex drawings of graphs in two and three dimensions. In *Proc. 12th Annu. ACM Sympos. Comput. Geom.*, pages 319–328, 1996.

[114] J. Cohen, A. Varshney, D. Manocha, G. Turk, H. Weber, P. Agarwal, F. P. Brooks, Jr., and W. V. Wright. Simplification envelopes. In *Proceedings of SIGGRAPH '96 (New Orleans, LA, August 4–9, 1996)*, Computer Graphics Proceedings, Annual Conference Series, pages 119–128. ACM SIGGRAPH, ACM Press, Aug. 1996.

[115] R. F. Cohen and R. Tamassia. Combine and conquer. *Algorithmica*, 18:342–362, 1997.

[116] G. Das and M. T. Goodrich. On the complexity of optimization problems for 3-dimensional convex polyhedra and decision trees. *Comput. Geom. Theory Appl.*, 8:123–137, 1997.

[117] O. Devillers, G. Liotta, F. P. Preparata, and R. Tamassia. Checking the convexity of polytopes and the planarity of subdivisions. *Comput. Geom. Theory Appl.*, 11(3–4):187–208, 1998.

[118] O. Devillers and F. P. Preparata. A probabilistic analysis of the power of arithmetic filters. *Discrete Comput. Geom.*, 20:523–547, 1998.

[119] O. Devillers and F. P. Preparata. Further results on arithmetic filters for geometric predicates. *Comput. Geom. Theory Appl.*, 13:141–148, 1999.

[120] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing.* Prentice Hall, Upper Saddle River, NJ, 1999.

[121] G. Di Battista, A. Garg, G. Liotta, A. Parise, R. Tamassia, E. Tassinari, F. Vargiu, and L. Vismara. Drawing directed acyclic graphs: An experimental study. *Internat. J. Comput. Geom. Appl.*, 10(6):623–648, 2000.

[122] G. Di Battista, R. Tamassia, and L. Vismara. Output-sensitive reporting of disjoint paths. *Algorithmica*, 23(4):302–340, 1999.

[123] G. Di Battista, R. Tamassia, and L. Vismara. Incremental convex planarity testing. *Information and Computation*, 166:1–33, 2001.

[124] M. Dickerson, C. A. Duncan, and M. T. Goodrich. K-D trees are better when cut on the longest side. In *Proc. 8th European Symp. on Algorithms*, volume 1879 of *Lecture Notes Comput. Sci.*, pages 179–190. Springer-Verlag, 2000.

[125] C. Duncan, M. T. Goodrich, and E. Ramos. Efficient approximation and optimization algorithms for computational metrology. In *ACM-SIAM Symp. on Discrete Algorithms*, pages 121–130, 1997.

[126] C. A. Duncan, M. T. Goodrich, and S. Kobourov. Balanced aspect ratio trees: combining the advantages of k-d trees and octrees. In *Proc. 10th Annu. ACM-SIAM Sympos. Discrete Alg.*, pages 300–309, 1999.

[127] C. A. Duncan, M. T. Goodrich, and S. G. Kobourov. Planarity-preserving clustering and embedding for large planar graphs. In *Proc. Graph Drawing*, volume 1731 of *Lecture Notes in Computer Science*, pages 186–196, 1999.

[128] C. A. Duncan, M. T. Goodrich, and S. G. Kobourov. Balanced aspect ratio trees and their use for drawing very large graphs. *Journal of Graph Algorithms and Applications*, 4(3):19–46, 2000.

[129] J. Erickson and D. Eppstein. Raising roofs, crashing cycles, and playing pool: Applications of a data structure for maintaining closest pairs. In *Proc. ACM Symp. on Computational Geometry (To appear)*, 1998.

[130] J. Erickson, L. J. Guibas, J. Stolfi, and L. Zhang. Separation sensitive collision detection for convex objects. In *Proc. 10th ACM-SIAM Sympos. on Discrete Algorithms*, pages 327–336, 1999.

[131] P. Gajer, M. T. Goodrich, and S. G. Kobourov. A fast multi-dimensional algorithm for drawing large graphs. In *Graph Drawing*, Lecture Notes in Computer Science, pages 211–221. Springer-Verlag, 2000.

[132] A. Garg, M. T. Goodrich, and R. Tamassia. Planar upward tree drawings with optimal area. *Internat. J. Comput. Geom. Appl.*, 6:333–356, 1996.

[133] A. Garg and R. Tamassia. GIOTTO3D: A system for visualizing hierarchical structures in 3D. In S. North, editor, *Graph Drawing (Proc. GD '96)*, Lecture Notes Comput. Sci. Springer-Verlag, 1997.

[134] A. Garg and R. Tamassia. On the computational complexity of upward and rectilinear planarity testing. *SIAM J. Computing*, 31(2):601–625, 2001.

[135] A. Garg, R. Tamassia, and P. Vocca. Drawing with colors. In *Proc. 4th Annu. European Sympos. Algorithms*, volume 1136 of *Lecture Notes Comput. Sci.*, pages 12–26. Springer-Verlag, 1996.

[136] N. Gelfand, M. T. Goodrich, and R. Tamassia. Teaching data structure design patterns. In *Proc. ACM Tech. Symp. on Computer Science Education (SIGCSE)*, 1998.

[137] N. Gelfand and R. Tamassia. Algorithmic patterns for orthogonal graph drawing. In S. H. Whitesides, editor, *Graph Drawing (Proc. GD '98)*, volume 1547 of *Lecture Notes Comput. Sci.*, pages 138–152. Springer-Verlag, 1998.

[138] M. Ghouse and M. T. Goodrich. Fast randomized parallel methods for planar convex hull construction. *Computational Geometry: Theory and Applications*, 7:219–235, 1997.

[139] G. A. Gibson, J. S. Vitter, and J. Wilkes. Report of the working group on storage I/O issues in large-scale computing. *ACM Computing Surveys*, 28(4):779–793, December 1996.

[140] M. T. Goodrich. Parallel algorithms in geometry. In J. Goodman and J. O'Rourke, editors, *CRC Handbook of Discrete and Computational Geometry*, pages 669–682. CRC Press, 1997.

[141] M. T. Goodrich. Randomized fully-scalable bsp techniques for multi-searching and convex hull construction. In *ACM-SIAM Symp. on Discrete Algorithms*, pages 767–776, 1997.

[142] M. T. Goodrich. Competitive tree-structured dictionaries. In *11th ACM-SIAM Symp. on Discrete Algorithms*, pages 494–495, 2000.

[143] M. T. Goodrich, L. Guibas, J. Hershberger, and P. Tanenbaum. Snap rounding line segments efficiently in two and three dimensions. In *ACM Symp. on Computational Geometry*, pages 284–293, 1997.

[144] M. T. Goodrich, M. Handy, B. Hudson, and R. Tamassia. Abstracting positional information in data structures: Locators and positions in JDSL. In *OOPSLA '98 Technical Notes*, 1998.

[145] M. T. Goodrich, M. Handy, B. Hudson, and R. Tamassia. Accessing the internal organization of data structures in the JDSL library. In M. T. Goodrich and C. C. McGeoch, editors, *Proc. Workshop on Algorithm Engineering and Experimentation*, volume 1619 of *Lecture Notes Comput. Sci.*, pages 124–139. Springer-Verlag, 1999.

[146] M. T. Goodrich, J. Mitchell, and M. W. Orletsky. Methods for approximate geometric pattern matching under rigid motion. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(4):371–379, 1999.

[147] M. T. Goodrich, M. Orletsky, and K. Ramaiyer. Methods for achieving fast query times in point location data structures. In *ACM-SIAM Symp. on Discrete Algorithms*, pages 757–766, 1997.

[148] M. T. Goodrich and K. Ramaiyer. Geometric data structures. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 463–489. Elsevier Science Publishers B.V. North-Holland, Amsterdam, 2000.

[149] M. T. Goodrich and E. Ramos. Bounded-independence derandomization of geometric partitioning with applications to parallel fixed-dimensional linear programming. *Discrete & Computational Geometry*, 18:397–420, 1997.

[150] M. T. Goodrich and R. Tamassia. Dynamic ray shooting and shortest paths via balanced geodesic triangulations. *J. Algorithms*, 23:51–73, 1997.

[151] M. T. Goodrich and R. Tamassia. Teaching the analysis of algorithms with visual proofs. In *Proc. ACM Symp. Computer Science Education*, 1998.

[152] M. T. Goodrich and R. Tamassia. Dynamic trees and dynamic point location. *SIAM Journal on Computing*, 28(2):612–636, 1999.

[153] M. T. Goodrich and C. G. Wagner. A framework for drawing planar graphs with curves and polylines. *J. Algorithms*, 37:399–421, 2000.

[154] G. Kant, G. Liotta, R. Tamassia, and I. G. Tollis. Area requirement of visibility representations of trees. *Inform. Process. Lett.*, 62(2):81–88, 1997.

[155] S. Kosaraju and G. Manzini. Compression of low entropy strings with lempel-ziv algorithms. In *Proc. SEQUENCES '97*, 1997.

[156] S. Kosaraju, A. Schaffer, and L. Biesecker. Approximation algorithms for a genetic dignostic problem. In *Workshop on Algorithms and Data Structures (WADS)*, volume 1272 of *LNCS*, pages 69–92, 1997.

[157] S. Kumar, D. Manocha, W. Garrett, and M. Lin. Hierarchical back-face culling. *Computers and Graphics*, 23(5):681–692, 1999.

[158] W. Lenhart and G. Liotta. Proximity drawings of outerplanar graphs. In S. North, editor, *Graph Drawing (Proc. GD '96)*, volume 1190 of *Lecture Notes Comput. Sci.*, pages 286–302. Springer-Verlag, 1997.

[159] G. Liotta, F. P. Preparata, and R. Tamassia. Robust proximity queries: an illustration of degree-driven algorithm design. *SIAM J. Comput.*, 28(3):864–889, 1998.

[160] G. Liotta, R. Tamassia, I. G. Tollis, and P. Vocca. Area requirement of Gabriel drawings. In *Algorithms and Complexity (Proc. CIAC' 97)*, volume 1203 of *Lecture Notes Comput. Sci.*, pages 135–146. Springer-Verlag, 1997.

[161] Y. Matias, E. Segal, and J. S. Vitter. Efficient bundle sorting. In *Proceedings of the 11th ACM-SIAM Symposium on Discrete Algorithms*, pages 839–848, San Francisco, January 2000.

[162] Y. Matias, J. S. Vitter, and M. Wang. Dynamic maintenance of wavelet-based histograms. In *Proceedings of the 2000 International Conference on Very Large Databases*, Cairo, September 2000.

[163] M. Pizzonia and R. Tamassia. Minimum depth graph embedding. In *Proc. European Symposium on Algorithms (ESA 2000)*, Lecture Notes in Computer Science. Springer-Verlag, 2000.

[164] R. Tamassia. Constraints in graph drawing algorithms. *Constraints*, 3(1):89–122, 1998.

[165] R. Tamassia. Implementing algorithms and data structures: an educational and research perspective. In *Proc. Annu. Internat. Sympos. Algorithms Comput.*, Lect. Not. Comput. Sci. Springer-Verlag, 1998.

[166] R. Tamassia. Advances in the theory and practice of graph drawing. *Theoret. Comput. Sci.*, 217:235–254, 1999.

[167] R. Tamassia. Graph drawing and information visualization. In *Proc. VIII Meetings on Computational Geometry (EGC8)*, 1999.

[168] R. Tamassia. Graph drawing. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 937–971. Elsevier Science Publishers B.V. North-Holland, Amsterdam, 2000.

[169] R. Tamassia, P. Agarwal, N. Amato, D. Chen, D. Dobkin, R. Drysdale, S. Fortune, M. Goodrich, J. Hershberger, J. O'Rourke, F. P. Preparata, J.-R. Sack, S. Suri, I. Tollis, J. Vitter, and S. Whitesides. Strategic directions in computational geometry. *ACM Comput. Surv.*, 28(4):591–606, 1996.

[170] R. Tamassia, M. T. Goodrich, L. Vismara, M. Handy, G. Shubina, R. Cohen, B. Hudson, R. S. Baker, N. Gelfand, and U. Brandes. JDSL: The data structures library in Java. *Dr. Dobb's Journal*, 323, April 2001.

[171] R. Tamassia, I. G. Tollis, and J. S. Vitter. A parallel algorithm for planar orthogonal grid drawings. *Parallel Processing Letters*, 10(1):141–150, March 2000.

[172] R. Tamassia and L. Vismara. A case study in algorithm engineering for geometric computing. *Internat. J. Comput. Geom. Appl.*, 11(1):15–70, 2001.

[173] K. Varadarajan. A divide and conquer algorithm for min-cost perfect matching in the plane. In *Proc. 39th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 320–329, 1998.

[174] K. Varadarajan and P. K. Agarwal. Approximating shortest paths on a nonconvex polyhedron. In *Proc. IEEE Symp. on Foundations of Comp. Sci.*, 1997.

[175] K. R. Varadarajan and P. K. Agarwal. Approximation algorithms for bipartite and non-bipartite matching in the plane. In *Proc. 10th ACM-SIAM Sympos. Discrete Algorithms*, pages 805–814, 1999.

[176] D. E. Vengroff and J. S. Vitter. Efficient 3-d range searching in external memory. In *Proceedings of the ACM Symposium on Theory of Computing*, volume 28, pages 192–201, Philadelphia, PA, May 1996.

[177] D. E. Vengroff and J. S. Vitter. I/O-efficient scientific computation using TPIE. In *Proceedings of NASA Goddard Conference on Mass Storage Systems*, volume 5, pages II, 553–570, September 1996.

[178] L. Vismara, G. Di Battista, A. Garg, G. Liotta, R. Tamassia, and F. Vargiu. Experimental studies on graph drawing algorithms. *Software Practice and Experience*, 30:1235–1284, 2000.

[179] J. S. Vitter. External memory algorithms and data structures: Dealing with MASSIVE data. *ACM Computing Surveys*, 33(2):209–271, June 2001. Available electronically via the author's web page `http://www.cs.duke.edu/~jsv/`.

[180] J. S. Vitter. External memory algorithms. In *Handbook of Massive Data Sets*. Kluwer Academic Publishers, in press.

[181] J. S. Vitter and M. Wang. Approximate computation of multidimensional aggregates of sparse data using wavelets. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Philadelphia, June 1999.

[182] H. Wang and P. K. Agarwal. Approximation algorithms for curvature constrained shortest paths. In *Proc. 7th ACM-SIAM Sympos. Discrete Algorithms*, pages 409–418, 1996.

3 Scientific Progress and Accomplishments

3.1 Introduction and Overview

Geometric computing, which emerged as a self-standing discipline about a quarter of a century ago under the broad rubric of Computational Geometry, was strongly motivated by the pervasive relevance of geometry to a large variety of applied fields, and was originally viewed as an exciting topic in the design and analysis of algorithms. In the context of this project, we underscore that Computational Geometry is a powerful aid to a number of military and civilian applications involving a physical or visual environment, including robotics, navigation, automatic target recognition (ATR), command, control, and communications (C3), battlefield management, geographic information systems, computational metrology, hit avoidance, computer-aided design, and computer graphics.

Over the years researchers in computational geometry developed an impressive corpus of algorithms and algorithmic paradigms, but became also aware of the severe limitations of the original model based on the assumed availability of exact real-number arithmetic. Instead, the reality of computing, which results in nonrobust implementations of mathematically correct algorithms, strongly hampered the projected transfer of technology that was the fundamental motivation of the field.

As a consequence of this unsatisfactory relation to real-world applications, in the last decade the field underwent a significant philosophical transformation, which positioned it strategically for the materialization of the desired connection with the applied areas. Specifically, a careful study of robust geometric computing and the serious development of readily available libraries of reliable geometric applications were the two most significant components of the proposal for the present project, which was submitted in response to the ARO MURI-initiative in early 1995. Funding for the five-year undertaking was granted in the summer of 1995.

At the completion of the funded research, we can confidently say that our coordinated efforts have contributed to the strengthening of the discipline and to the forging of an effective partnership between academic research and practice, both civilian and military. In fact, this re-orientation of the research community [169] has been met by a complementary acceptance by the application community, which overcame its initial skepticism and recognized the invaluable power of geometric computing, because the latter is grounded on solid algorithmic and topological knowledge and uses the most advanced computing technologies.

In summary, we believe that our work has helped prepare a most fertile ground for the further evolution of the field in a rapidly changing technological landscape. Modern computers have processors and graphic engines that are fast enough to perform significant computations on collections of 2- and 3-dimensional geometric objects, such as points, line segments, polygons, curves, surfaces, etc. In addition, the advent of the Internet has brought about new paradigms and modalities for using computers to solve large problems, which are often geometric. Moreover, these paradigms and modalities often involve interesting connections to other areas of computer science, including distributed computing, computer science education, and electronic commerce.

Research initiatives and detailed individual results the previous five interim reports, duly generated at the conclusion of each year of funded research. In this final report, however,

along with the major thematic highlights of the project, we emphasize our most recent research.

In a nutshell, the guiding principles of our research have been the pursuit of algorithmic robustness and ease-of-programming, as well as the effective management of scale. These are the features of algorithm engineering, which is key to the targeted transfer of technology. Experimental verification was an important component of this effort as well.

Our presentation is organized as follows. In Section 3.2, "Foundations", we review our contributions to basic algorithmic research in computational geometry, such as algorithms pertaining to geometric structures of paramount significance (such as arrangements) and general data organization. In Section 3.3, "Methodologies", we summarize our contributions to correcting the model inadequacies of traditional computational geometry, specifically:

- methods for the design of practical and robust geometric algorithms,
- development of a finer performance framework than asymptotic analysis,
- external memory management techniques for large-scale geometric computations,
- a transparent parallel I/O environment for large-scale geometric computations (TPIE),
- techniques for effectively integrating I/O systems into massively parallel computers.

In Section 3.4, "Emerging Technologies", we review the adaptation of our research effort to the new computing realities, principally new powerful software-engineering paradigms and the Internet:

- an object-oriented library of robust geometric algorithms (GeomLib),
- development of a distributed architecture for geometric computing over the Web (GeomNet),
- development of approximation algorithms for geometric optimization problems,
- algorithms and data structures for mobile data.

Finally, in Section 3.5, "Applications", we discuss our numerous contributions to a variety of applied areas.

3.2 Foundations

Given our emphasis on understanding the fundamental aspects of geometric computing, we studied in 1999 several fundamental problems in combinatorics and in computational geometry. We show in [76], using several methods, that there exists a set S of n points in the d-dimensional unit cube so that every $d+1$ points of S define a simplex of volume $\Omega(\frac{1}{n^d})$.

In [82] we define a new type of distance functions in the plane from a point to a *pair* of points. We focus on a few such distance functions, analyze the structure and complexity of the corresponding nearest- and furthest-neighbor Voronoi diagrams (in which every region is defined by a pair of point sites), and show how to compute the diagrams efficiently.

A graph H is a universal graph for a family of graphs if every graph in the family is a subgraph of H. In [107] we construct a universal graph of size $O(n)$ for the family of n-vertex planar graphs of bounded degree. This is a significant improvement over the previously known bound of $O(n \log n)$. In [108] we construct a universal graph of size $O(n)$ for the family of n-vertex, bounded degree graphs with a bisector of size $O(n^c)$, for some $c < 1$. This is a significant improvement over the previously known bound of $O(n^{2c})$.

3.2.1 Arrangements of surfaces

We studied several problems involving arrangements of surfaces. In [15, 10], we considered the problem of bounding the complexity of the k-th level in an arrangement of n curves or surfaces, a problem dual to, and an extension of, the well-known *k-set problem.* Among other results, we prove a new bound, $O(nk^{5/3})$, on the complexity of the k-th level in an arrangement of n planes in R^3, or on the number of k-sets in a set of n points in three dimensions, and we show that the complexity of the k-th level in an arrangement of n line segments in the plane is $O(n\sqrt{k}\alpha(n/k))$, and that the complexity of the k-th level in an arrangement of n triangles in 3-space is $O(n^2k^{5/6}\alpha(n/k))$.

In another result, we proved a subcubic bound on the complexity of the union of n congruent cylinders. This is the first step toward proving a near-optimal bound on the complexity of the Voronoi diagram of lines or of polygons in 3-space.

In [18], we obtained improved bounds on the complexity of many faces in arrangements of circles and unit circles. In [52], we extended the duality transform to a family of pseudo-lines (a family of curves, any two of which intersect in at most one point), described an algorithm for computing this duality, and showed how this transform can be used to develop efficient algorithms for a number of problems involving circles and other quadratic curves. We wrote two comprehensive surveys on this topic [49, 48].

3.2.2 Geometric Algorithms and Data Structures

We give very fast work-efficient parallel algorithms [138] for finding convex hulls of planar point sets and solving fixed-dimensional linear programs [149] in a computational environment that allows for fine-grain parallelism. Our work for linear programming actually gave a general method for efficiently derandomizing randomized geometric algorithms sequentially or in parallel and uses linear programming as a motivating example. We also give optimal coarse-grain parallel algorithms [141] for convex hull construction and a general class of data structure searching problems. We review and survey [140] in an invited book chapter the major results in parallel computational geometry, many of which are the results of Goodrich, Preparata, Tamassia, and Vitter.

We define and study a combinatorial problem [156] called the Weighted Diagnostic Cover (WDC), which models genotyping in the diagnosis of a class of chromosomal aberrations. We develop several approximation algorithms for WDC. We establish worst-case performance bounds for these algorithms. We also report their performance on a real data set.

In [24], we developed output-sensitive algorithms for reporting all pairs of intersecting polytopes in a given set of convex polytopes in 3-space. This is the first truly output-sensitive algorithm for the problem.

Motivated by the fact that geometric computing naturally involves large volumes of pictorial images, and considering the high demand of raw pictures both in transmission over the net and in local storage, we have thought it appropriate to revisit the effectiveness of competing data compression techniques. We have compared [155] the compression ratio of the Lempel-Ziv algorithms with the empirical entropy of the input string. We show that although these algorithms are optimal according to the generally accepted definition, we can find families of low entropy strings which are not compressed optimally. We then present a compression algorithm which combines Lempel-Ziv with Run Length Encoding, and we show that it has excellent performance.

In [148] we describe a host of geometric data structures and how they can be used to efficiently store and query geometric information. In [124] we show that the classic k-D tree data structure actually has efficient polylogarithmic worst-case time performance for answering approximate nearest-neighbor searches, which confirms in a theoretical analysis a behavior that researchers have often observed in practice.

In [22], we developed the first algorithm for constructing an R-tree, a widely used data structure in practice, with provable near-optimal query bounds. In [8] we developed the data structure for bulk-loading kd-trees and for performing updates. We wrote two comprehensive surveys on range searching [1, 28].

In [59] we describe an efficient algorithm for computing arrangements of curve segments. In [60] we solve a long-standing open problem in computational geometry, showing a simple randomized linear-time algorithm for triangulating an n-vertex simple polygon.

3.2.3 Clustering problems

Clustering a set of points into a few groups is frequently used for statistical analysis and classification in numerous applications, including information retrieval, facility location, data mining, spatial data bases, data compression, image processing, astrophysics, and scientific computing.

In [43] we presented an $n^{O(k^{1-1/d})}$-time algorithm for solving the k-center problem in R^d, under any L_p metric. We also described a simple $(1+\epsilon)$-approximation algorithm for the k-center problem, with running time $O(n \log k) + (k/\epsilon)^{O(k^{1-1/d})}$. We also presented an $n^{O(k^{1-1/d})}$-time algorithm for solving the L-capacitated k-center problem, provided that $L = \Omega(n/k^{1-1/d})$ or $L = O(1)$.

In [53], we presented an $O(n^{4/3} \log^5 n)$-time algorithm for computing the *discrete 2-center* of a set P of n points in the plane; that is, computing two congruent disks of smallest possible radius, centered at two points of P, whose union covers P.

One method to reduce the dimensionality is *projective clustering*: Given a set S of n points in a d-dimensional space and two integers $k < n$ and $q \leq d$, find k q-dimensional flats $h_1, \ldots, h_k$ and partition S into k subsets $S_1, \ldots, S_k$ so that $\max_{1 \leq i \leq k} \max_{p \in Si} d(p, h_i)$ is minimized. In [44], we consider the following two instances of the projective clustering problem: Given a set S of n points in and an integer $k > 0$, cover S by k hyper-strips (resp. hyper-cylinders) so that the maximum width of a hyper-strip (resp., the maximum diameter of a hyper-cylinder) is minimized. We presented efficient approximation algorithms in two and three dimensions. In 2D the running time is almost linear as a function of the number of points. Recently, we developed efficient ϵ-approximation algorithms for projective clustering

in 2D. In [57] we developed and implemented practical algorithms for projective clustering and applied them to indexing images.

The papers [45, 46] survey known geometric algorithms for clustering and other geometric optimization problems.

3.2.4 Data Organization and Information Visualization

The summarization, storage, and visualization of large amounts of data plays a major role in knowledge mining and decision support systems. We are investigating techniques for the visualization of relational and quantitative information, and of their interplay. Target applications include Web search engines, command and control, and programming environments. In [126] we describe a new geometric data structure for clustering a set of points so as to facilitate fast data mining on that set. The data structure, which we call the balanced aspect ratio (BAR) tree, combines the best features of octrees and k-d trees, in that it achieves $O(\log n)$ height, has $O(n)$ space, and defines cluster regions with bounded aspect ratio. Such a structure can be used, for example, to answer approximate nearest-neighbor queries and approximate range searching queries in $O(\log n)$ time, and our experimental analysis shows that they run fast in practice as well.

In [152] we show how to maintain a planar subdivision (such as defined by a Voronoi diagram) so as to quickly support point location queries as well as dynamic updates to the subdivision. Our data structure achieved $O(\log n)$ time updates and $O(\log^2 n)$ time queries. A general technique for dynamizing data structures used in geometric searching and graph drawing is presented in [115]. Optimal data structures for shortest path and minimum-link path queries between two convex polygons inside a simple polygonal obstacle are presented in [112].

Current work on information visualization has focused on algorithms and systems for graph drawing. In [127] we present a novel approach for cluster-based drawing of large planar graphs that maintains planarity. Our technique works for arbitrary planar graphs and produces a clustering which satisfies the conditions for compound-planarity (c-planarity). Using the clustering, we obtain a representation of the graph as a collection of $O(\log n)$ layers, where each succeeding layer represents the graph in an increasing level of detail. At the same time, the difference between two graphs on neighboring layers of the hierarchy is small, thus preserving the viewer's mental map. The overall running time of the algorithm is $O(n \log n)$, where n is the number of vertices of graph G.

In [110] we address the problem of drawing planar graphs with circular arcs while maintaining good angular resolution and small drawing area. We present a lower bound on the area of drawings in which edges are drawn using exactly one circular arc. We also give an algorithm for drawing n-vertex planar graphs such that the edges are sequences of two continuous circular arcs. The algorithm runs in $O(n)$ time and embeds the graph on the $O(n) \times O(n)$ grid, while maintaining $\Theta(1/d(v))$ angular resolution, where $d(v)$ is the degree of vertex v. Since in this case we use circular arcs of infinite radius, this is also the first algorithm to simultaneously achieve good angular resolution, small area and at most one bend per edge using straight-line segments. Finally, we show how to create drawings in which edges are smooth C^1-continuous curves, represented by a sequence of at most three circular arcs.

In [128] we develop an efficient algorithm for displaying large graphs. In [131] we give a fast method for drawing large graphs using force-directed methods. In [153] we describe a linear-time algorithm for drawing n-node planar graphs in an $O(n) \times O(n)$ grid with few bends per edge.

In [106] we describe a fast data structure for performing range queries in tree cross products. In [142] we describe a new data structure for ordered dictionaries, which we call the energy-balanced trees. These trees achieve logarithmic amortized update times and logarithmic worst-case search time, but use only partial rebuilding to achieve balance; they do not use rotations.

3.3 Methodologies

As mentioned above, earlier research on geometric computing referred to a convenient computation model, borrowed from traditional algorithmic research and based on the assumptions of unbounded internal memory and exact real-number arithmetics. This section described our efforts at correcting the arising shortcomings.

3.3.1 Algorithmic Robustness

Robustness has been a central issue throughout our project.

In [159] we have elaborated on the exact computation paradigm and formalized the notion of degree of a geometric algorithm, as a worst-case quantification of the precision (number of bits) to which arithmetic calculation have to be executed in order to guarantee topological correctness. Such criterion characterizes the computational cost of topological correctness of the output for the case of noiseless input data, and therefore spells out the degree as an important parameter of algorithmic design. Of course, concrete situations are considerably more complicated. First of all, input data may be noisy, i.e., affected by errors inherent in physical measurements or expressing the limited accuracy of previous computations. In such cases, the topology consistent with the data is, in general, not unique (depending upon the magnitude of the noise), and the algorithm must produce its own certificate (multiple topologies, or a topology consistent with specified nominal values). Second, and certainly not less important, while the algorithmic degree specifies the worst-case arithmetic demands, a substantial majority of the computations can be confidently executed on a floating-point platform, acting on floating-point approximations of the underlying integer operands. Here again, extreme care must be exercised in assuring the validity of the approximation. This task is known as the design of arithmetic filters, which we are actively investigating, following our initial evaluation of some important geometric tests [118, 119]. A main direction is the development of tight interval analysis for the very concrete case of noisy input data.

In the degree-driven algorithmic design area, we have successfully redesigned algorithm for robust proximity queries [159] and have revisited [97] the plane-sweep algorithms for intersecting segments. This is a fundamental application, but the available algorithms are notorious for not being robust. Our analysis shows that the reported unrobustness is attributable to the deployment of arithmetics inadequate for the specified calculations, and we present algorithms that either match or closely approach the lower bounds on the arithmetic and are eminently practical, with no sacrifice of efficiency.

In [143], we provide an improved algorithm for performing snap rounding of an arrangement of line segments in the plane, whose techniques conform with the criterion of low-degree design, outlined above.

3.3.2 External memory algorithms

The data sets for many of today's computer applications are too large to fit within the computer's internal memory and must instead be stored on external storage devices such as disks. A major performance bottleneck can be the input/output communication (or I/O) between the external and internal memories. We have written the first survey articles on the field [179, 180, 139], in which we give a detailed overview of the state of the art.

Much of our work on external memory algorithms has concentrated on geometric problems with immediate applications in spatial databases and geographic information systems (GIS). Modern GIS systems, and especially military systems, are often very data-intensive and thus they require good use of external memory techniques. We discuss GIS systems further in Section 3.5.1.

During the year we have intensified our work on implementing our external memory algorithms in the TPIE system [177]. TPIE (Transparent Parallel I/O Environment) is designed to allow programmers to write applications that make efficient use of I/O resources. In [65] we thus not only developed a powerful theoretical framework for designing efficient algorithms for large-scale batched dynamic problems, but we also implemented one of our algorithms in TPIE and showed that for large problem sizes it greatly outperforms other well-known algorithms. We have also continued this work and showed that our TPIE implementation of a theoretically developed algorithm for the extremely important spatial database problem called *spatial join* performs as well as the state-of-the-art algorithm developed in the spatial database community. Unlike previously known implementations our implementation also performs well on skewed or higher dimensional data. We have started our previously proposed extension of TPIE to include commonly used external data structures and are very encouraged by the results obtained so far. We have implemented a version of the so called R-trees—the data structure which has emerged as an efficient and effective indexing method for spatial data—and demonstrated how the time spent on construction such structures can be dramatically improved using external memory techniques [63].

Many data sets to be sorted consist of a limited number of distinct keys. Sorting such data sets can be thought of as bundling together identical keys and having the bundles placed in order; we therefore denote this as *bundle sorting*. In [161], we described an efficient algorithm for bundle sorting in external memory. We show that our algorithm is optimal by proving a matching lower bound for bundle sorting. The improved running time of bundle sorting over general sorting can be significant in practice, as demonstrated by experimentation. An important feature of the new algorithm is that it is executed "in-place", requiring no additional disk space.

Computing multidimensional aggregates in high dimensions is a performance bottleneck for many OLAP applications. Obtaining the exact answer to an aggregation query can be prohibitively expensive in terms of time and/or storage space in a data warehouse environment. It is advantageous to have fast, approximate answers to OLAP aggregation queries. In [181], we presented a novel method based on wavelets that provides approximate answers

to high-dimensional OLAP aggregation queries in massive sparse data sets in a time-efficient and space-efficient manner. In [162], we consider the case in which the data are updated dynamically on an ongoing basis. Experiments on real data show that our method provides significantly more accurate results for typical OLAP aggregation queries than other efficient approximation techniques such as random sampling.

In [66], we settled several longstanding open problems in theory of indexability and external orthogonal range searching. In the first part of the paper, we apply the theory of indexability to the problem of two-dimensional range searching. We show that the special case of 3-sided querying can be solved with constant redundancy and access overhead. From this, we derive indexing schemes for general 4-sided range queries that exhibit an optimal tradeoff between redundancy and access overhead. In [69], we derived algorithms with similar performance for the special case of stabbing queries, which is needed for dynamic interval maintenance.

In [176], we present a new approach to designing data structures for the important problem of external-memory range searching in two and three dimensions. We construct data structures for answering range queries in $O((\log\log\log_B N)\log_B N + K/B)$ I/O operations, where K is the output size. We base our data structures on the novel concept of B-approximate boundaries, which are manifolds that partition space into regions based on the output size of queries at points within the space.

We consider the problem of devising external memory algorithms whose memory allocations can change dynamically and unpredictably at run-time. In [94], we presented a simple and natural dynamic memory allocation model. We define memory-adaptive external memory algorithms and specify what is needed for them to be dynamically optimal. Using novel techniques, we design and analyze dynamically optimal memory-adaptive algorithms for the problems of sorting, permuting, FFT, permutation networks, (standard) matrix multiplication and LU decomposition. The lower bound proof techniques for sorting and matrix multiplication are fundamentally distinct techniques, and they are invoked by most other external memory lower bounds; hence we anticipate that the techniques presented here will apply to many external memory problems.

In [92], we consider a cache shared by several concurrently running application processes and propose a provably efficient application-controlled global strategy for the shared cache. Using future information implicitly in the form of good decisions by application processes, we are able to break through the H_k lower bound on competitive ratio proved for classical paging for a k-sized cache. For a size-k cache shared by P application processes that always make good cache replacement decisions, we develop an online application-controlled paging algorithm with a competitive ratio of $2H_{P-1} + 2$. Typically, P is much smaller than k, perhaps by several orders of magnitude. Our competitive ratio improves upon the $2P + 2$ competitive ratio achieved by Cao et al. We show for this problem that no online algorithm A can have a competitive ratio better than H_{P-1} even if the application processes aiding A have perfect knowledge of individual request sequences. Our results are with respect to a worst-case interleaving of the individual request sequences of the P applications. We also consider other realistic notions of fairness.

In [93], we provide a competitive analysis framework for online prefetching and buffer management algorithms in parallel I/O systems, using a read-once model of block references. This has widespread applicability to key I/O-bound applications such as external merging

and concurrent playback of multiple video streams. Two realistic lookahead models, global lookahead and local lookahead, are defined. Algorithms NOM and GREED based on these two forms of lookahead are analyzed for shared buffer and distributed buffer configurations, both of which occur frequently in existing systems. An important aspect of our work is that we show how to implement both the models of lookahead in practice using the simple techniques of forecasting and flushing.

In [6], we show how to preprocess a set S of points in d-dimensional Euclidean space to get an external memory data structure that efficiently supports linear-constraint queries. Each query is in the form of a linear constraint $\mathbf{a} \cdot \mathbf{x} \leq \mathbf{b}$; the data structure must report all the points of S that satisfy the query. (This problem is called Halfspace range searching in the computational geometry literature.) Our goal is to minimize the number of disk blocks required to store the data structure and the number of disk accesses (I/Os) required to answer a query. For $d = 2$, we present the first near-linear size data structures that can answer linear-constraint queries using an optimal number of I/Os. We also present a linear-size data structure that can answer queries efficiently in the worst case. We combine these two approaches to obtain tradeoffs between space and query time. Finally, we show that some of our techniques extend to higher dimensions.

3.4 Emerging Technologies

3.4.1 Kinetic Data Structures

In applications like virtual reality and dynamic simulations, specified input objects move or deform continuously. In this context, the questions of computing some attribute is replaced by that of maintaining it over time. We studied several problems involving motion using the approach of kinetic data structures (KDS) developed at Stanford. Most of the work has been done in collaboration with the MURI center at Stanford.

A binary space partition is an essential structure used in particular in visualization. Its maintenance over time as objects move is important for interactive simulations. In [34], we describe the first known algorithm for efficiently maintaining a Binary Space Partition (BSP) for n continuously moving segments in the plane. Under reasonable assumptions on the motion, we show that the total number of times the BSP changes is $O(n^2)$, and that we can update the BSP in $O(\log n)$ expected time per change. We also consider the problem of constructing a BSP for n triangles in three-dimensional Euclidean space. We present a randomized algorithm that constructs a BSP of expected size $O(n^2)$ in $O(n^2 \log^2 n)$ expected time. We also describe a deterministic algorithm that constructs a BSP of size $O((n+k)\log n)$ and height $O(\log n)$ in $O((n+k)\log^2 n)$ time, where k is the number of intersection points between the edges of the projections of the triangles onto the xy-plane. We later extended this result to maintaining a BSP of moving polyhedral objects [29]. We also proved lower bounds for kinetic BSPs and triangulations [19].

In [20, 130], we presented kinetic data structures for detecting collisions between a set of polygons that are not only moving continuously but whose shapes can also change continuously with time. We construct a planar subdivision of the common exterior of the polygons, called *pseudo-triangulation*, which certifies their disjointness. See also [129].

We applied the kinetic setting to a very different set of problems in parametric optimiza-

tion. Several questions that involve finding an optimum tradeoff between two quantities (e.g. cost and reliability) can be stated as parametric optimization problem. We studied the parametric minimum spanning tree problem and obtained substantial improvements over previous known solutions [27].

In [33] we developed a kinetic data structure for maintaining various extants of moving points, including diameter, width, and minimum bounding rectangle. It turns out that maintaining any of these extants exactly is expensive, so we recently developed considerably faster algorithms for maintaining them approximately [35]. In [5], we proposed various indexing schemes to store a set S of N points in the plane, each moving along a linear trajectory, so that a query of the following form can be answered quickly: Given a rectangle R and a real value t (called *time stamp*), report all points of S that lie inside R at time t. We propose algorithms that use kinetic data structures as well time-space framework. We obtain a tradeoff between the query time and the time spent in evolving the data structure as the points move. We also present an indexing scheme whose query time depends on the value of t, the time stamp of the query t. In [9] we extended these techniques to obtain a trade-off between query time and the time spent in updating the structure. In recent work, we developed a novel method for maintaining an R-tree on a set of moving points in the plane. An R-tree is a commonly used data structure for answering various geometric queries.

3.4.2 Elements of a Geometric Library (GeomLib)

The *GeomLib* project addresses the important objective of developing an easy to use, reliable, open library of robust and efficient geometric algorithms. Recent object-oriented design concepts such as design patterns and algorithm abstraction are extensively used throughout the entire project. In the design phase, we have taken into account the experience of other similar efforts, such as the Library of Efficient Data structures and Algorithms (LEDA) and the more recent Computational Geometry Algorithms Library (CGAL).

- To provide researchers in computational geometry with a framework for algorithm engineering, with a specific emphasis on geometric computing. In this context, GeomLib will be typically used for rapid prototyping and experimental studies of geometric algorithms.
- To make computational geometry results available to the users in other areas, such as robotics, geographic information systems (see Section 3.5.1), mechanical engineering, computer graphics, etc.

The principles, architecture, and design of GeomLib are presented in [172]. This paper also demonstrates the applicability of algorithm engineering and software design concepts to geometric computing through a vertical case study on the implementation of planar point location algorithms within GeomLib (see also http://www.jdsl.org/src/pointloc/). The design and implementation of the core data structures support for GEOMLIB is presented in [144, 145, 165]. Related work on checkers for verifying the correctness of geometric structures such as convex polytopes and planar subdivisions are presented in [117]. Related work on algorithm engineering and applications to computer science education is presented in [95, 136, 151].

A major milestone in the development of GEOMLIB has been the release of version 2.0 the combinatorial component of GEOMLIB under the name of JDSL in August 2000 [170] (see http://www.jdsl.org/). Six months after its release, JDSL 2.0 has been downloaded by more than 2,500 users worldwide. Also, more than 100 users have expressed interest in collaborating to the extension of the library.

3.4.3 Geometric Internet Computing (GeomNet)

In [78] we present *GeomNet*, a system for performing distributed geometric computing over the Internet. The mail goal of the system is facilitating software evaluation and adoption, as well as serving for many other purposes, through providing easy (Internet) access to a variety of complex geometric algorithms. We describe the architecture of GeomNet as a series of layers and discuss the connections between them. We also provide several examples of geometric algorithms that our system already supports. Application domains for GeomNet include collaborative research, distance education, and software marketing.

We have developed two prototypes of geometric Web computing services under GeomNet: the Mocha algorithm animation system [72, 71, 70, 73], and the Graph Drawing Server [100].

Mocha supports interactive animations over the Web of geometric algorithms. It employs a client-server architecture that optimally partitions the software components of a typical algorithm animation system, and leverages the power of the Java language. Mocha has high levels of security, protects the algorithm code, places a light communication load on the Internet, and allows users with limited computing resources to access animations of computationally expensive algorithms. The user interface combines fast responsiveness and user friendliness with hypertext narratives. The architecture of Mocha has been successfully adopted by researchers worldwide.

The Graph Drawing Server can be used for drawing graphs from user-applications, studying and comparing graph drawing algorithms, translating between the formats for describing graphs and their drawings, creating a database of graphs occurring in user-applications, an providing demonstrations in an educational setting. The service allows the user either to specify the graph in a formal way (HTML format) or to interactively draw the input graph on the screen.

3.5 Applications

3.5.1 Geographic Information Systems

Geographic information systems (GIS), which consist of spatial databases for storing geometric mapping information, offer important applications for our research. We have been collaborating on many geometric problems that arise in GIS with geometric, geographic, and GIS researchers worldwide, including those at the Nicholas School of Environment at Duke.

The potential and use of Geographic Information Systems (GIS) is rapidly increasing because of the increasing availability of massive amounts of geospatial data from projects like NASA's Earth Observing System and Earth Science Enterprise. However, the use of these massive datasets also exposes scalability problems with existing GIS algorithms. These scalability problems arise from the fact that most GIS algorithms have been designed to

minimize internal computation time, whereas I/O communication often is the bottleneck when processing massive amounts of data. In [67, 62, 61], we consider I/O-efficient algorithms for problems on grid-based terrains. Detailed grid-based terrain data is rapidly becoming available for much of the earth's surface. We demonstrate the practical merits of our work by comparing the empirical performance of our new algorithm for the *flow accumulation* problem with that of the previously best known algorithm. Our experiments show that while our new algorithm scales nicely with dataset size, the previously known algorithm "breaks down" once the size of the dataset becomes bigger than the available main memory.

Most spatial join algorithms either assume the existence of a spatial index structure that is traversed during the join process, or solve the problem by sorting, partitioning, or on-the-fly index construction. In [64], we developed a simple plane-sweeping algorithm that unifies the index-based and non-index based approaches. This algorithm processes indexed as well as non-indexed inputs, extends naturally to multi-way joins, and can be built easily from a few standard operations. We present the results of a comparative study of the new algorithm with several index-based and non-index based spatial join algorithms. We consider a number of factors, including the relative performance of CPU and disk, the quality of the spatial indexes, and the sizes of the input relations. An important conclusion from our work is that using an index-based approach whenever indexes are available does not always lead to the best execution time, and hence we propose the use of a simple cost model to decide when to follow an index-based approach.

We have studied a number of data structure engineering issues for storing and querying GIS data. We describe an adaptive scheme [147] for "morphing" a point-location GIS data structure so that it can adapt to non-uniform query distributions in a way that allows for faster than logarithmic-time response to some queries. We also provide an efficient algorithm [150] for performing ray shooting queries in two-dimensional planar subdivisions.

We have developed a new theoretical optimal algorithm for the important GIS problem of extracting contour line from terrain data stored as Triangulated Irregular Networks (TIN) [7]. We have also developed new disk space and query efficient—and sometimes even provably optimal—data structures for the problem of storing a set of points in R^d on disk such that linear-constraint queries can be answered efficiently [6]. A linear-constraint query consists of finding all points lying below a query hyperplane.

We developed an I/O efficient dynamic point-location data structure that uses linear space, supports fast insertion and deletion operations, and can answer point-location queries [4]. In [55] we developed approximation algorithms for labeling point features on maps, and in [39] we described several heuristics for labeling linear features (e.g. rivers, roads) in a map.

We considered the problem of computing the shortest path between two points on a given polyhedral surface. This is a central problem in numerous areas, including robotics, geographic information systems, medical imaging, low-altitude flight simulation, and water-flow analysis. In most of these applications where efficiency is critical, the input surfaces are large but they are still an approximation of the real surface. Hence, a simple, efficient algorithm for computing an approximate shortest path is preferable to an expensive algorithm that computes an exact shortest path. For convex surfaces, we obtained a linear time algorithm that computes a path that whose length is at most $1+\epsilon$ that of the shortest path [37]. For non-convex surfaces, We obtained the first algorithms that compute an approximate shortest path in subquadratic time [174]. In [36], we have developed and implemented a

robust, efficient algorithm for computing approximate shortest paths on a convex polyhedral surface. Although the analysis and the current implementation of the algorithm work only for convex polytopes, many of ideas extend to arbitrary polyhedral surfaces.

In [68], we develop efficient new external-memory algorithms for a number of important problems involving line segments in the plane, including trapezoid decomposition, batched planar point location, triangulation, red-blue line segment intersection reporting, and general line segment intersection reporting. In GIS systems, the first three problems are useful for rendering and modeling, and the latter two are frequently used for overlaying maps and extracting information from them.

3.5.2 Robotics

We consider the problem of computing the shortest path between two points on a given polyhedral surface. This is a central problem in numerous areas, including robotics, geographic information systems, medical imaging, low-altitude flight simulation, and water-flow analysis. In most of these applications, a simple, efficient algorithm for computing an approximate shortest path is preferable to an expensive algorithm that computes an exact shortest path, since the input surfaces are large, efficiency is critical, and the polyhedral surface is typically an approximation of the real surface. All previous approaches take superquadratic time to compute even an approximate shortest path. We present the first algorithms that compute an approximate shortest path in subquadratic time. An undergraduate student is currently implementing our algorithm.

When a robot is too big to be assimilated to a point, the next approximation is to consider the smallest ball containing it. If we wish to obtain a path for the ball that doesn't collide with any obstacle in space, a standard approach is to reduce this problem to that of planning the motion of a point in a space where each obstacle is grown by the radius of the ball. We show that this space has a complexity roughly quadratic in the number of objects [50, 47]. Only the trivial cubic bound was known previously.

We have also been studying the optimal motion-planning problems under nonholonomic constraints. We have developed a simple, efficient algorithm for computing a curvature-constrained approximate shortest path for a point robot moving amid planar obstacles [182]. We have also developed a simple, efficient algorithm for computing a curvature-constrained approximate shortest path for a point robot moving inside a convex polygon. We proved several interesting properties of shortest paths, which we believe are useful for navigation amid planar obstacles [21]. In collaboration with the robotics group at Stanford, we studied the problem of moving an object by pushing it [40]. See [2, 3, 16, 23, 30, 12, 32] for other robotics problems, including assembly design, penetration depth, and polygon placement, which we studied.

3.5.3 Information Visualization

The visualization of large amounts of data plays a major role in information retrieval, knowledge mining and decision support systems, which are critical applications especially in relation to the digital battlefield. In this context we have devoted particular attention to geometrically visualizing graph and networks and we have thus focused on graph drawing

algorithms and systems.

The way in which space is partitioned relative to a geometric model can also have a big impact on rendering speed. Thus, as another step towards developing efficient rendering algorithms, we have been studying *binary space partitions* (BSP), which is a data structure of fundamental importance in visualization, with applications in hidden-surface removal, ray tracing, global illumination, and shadow generation. All the lower-bound constructions for large-size BSPs occur very rarely in practice. Continuing our work on constructing BSPs for orthogonal rectangles [31, 41], we studied the problem of constructing BSPs for triangles [34]. Our algorithms perform very efficiently particularly when the triangles model terrain-like surfaces. With the emergence of VRML, many geometric data sets contain moving objects. Motivated by this fact, we have developed a fast algorithm for maintaining the BSP for continuously moving triangles, which we describe in Section 3.4.1.

In [98, 99], we address the visualization of large-scale transportation and communication networks and present a new edge routing method that uses cubic curves in the plane and on the sphere. We also provide novel interactive visualizations of air and train connections and of multicast Internet traffic. Experiments on real-world data sets indicate that our method is computationally fast and can effectively visualize networks with 7,000 nodes and 10,000 edges.

In [102], we introduce a framework for defining and validating metrics to measure the difference between two drawings of the same graph, and give a preliminary experimental analysis of several simple metrics. In [103], we formally define several intuitive ideas of similarity and present the results of a user study designed to evaluate how well these measures reflect human perception of similarity.

In [128] we develop an efficient algorithm for displaying large graphs using geometric space-partitioning techniques. In [131] we give a fast method for drawing large graphs using force-directed methods.

A systematic study of planar drawings has been conducted with the goal of optimizing the embedding of graphs in the plane. An optimal compaction method for orthogonal representations is given in [104]. Convex planar grid drawings are studied in [122, 123]. Planar tree drawings are studied in [109, 132, 154]. A parallel orthogonal drawing technique is presented in [171]. Bend and area minimization in curvilinear drawings is addressed in [153, 110]. In [163], we give a linear time algorithm for computing a minimum-depth embedding of a planar graph. A linear-time algorithm for testing the upward planarity of single-source digraphs is presented in [96]. In [134], we show that upward planarity testing and rectilinear planarity testing are NP-complete problems. In [127] we present a novel approach for cluster-based drawing of large planar graphs that maintains planarity.

A system for the interactive construction of orthogonal drawings is presented in [105]. An object-oriented design and a software prototype of a fundamental orthogonal graph drawing technique is presented in [137]

Three-dimensional drawings of graphs are studied in [113, 116, 133, 135].

In [158, 13, 121, 160], we explore various types of proximity drawings, characterize classes of graphs that admit such drawings, and investigate the time complexity of constructing them.

In [101], we present an application of our graph drawing and Internet computing techniques to computer science education.

The book [120] is recognized as the authoritative reference in the field of graph drawing and can be used as a textbook in graduate courses. A survey of research on graph drawing is given in [168]. Recent developments in graph drawing research are overviewed in [166] and [167]. In [178], we survey experimental studies on graph drawing algorithms. Techniques for supporting constraints in a wide variety of graph drawing algorithms are discussed in [164].

3.5.4 Geometric Modeling and Computer Graphics

Computer-aided geometric modeling and computer graphics are important topics to the Army because of the central role it plays in vehicle design, testing, and battle visualization and simulation. Of particular significance is the role that BRL-CAD plays in establishing the vulnerability of various military assets. We have therefore carried out several activities in these areas.

In [75] we describe a method for using geometric hashing to efficiently fix a CAD object whose boundary is broken due to imprecision, erroneous approximation of smooth surfaces, etc. The algorithm preserves global orientability of the repaired object. We continue and extend work on CAD repair in work that has been of significant interest, particularly to our colleagues at ARL-Aberdeen. Specifically, we have designed a general framework and new model-repair heuristic algorithm that couples polygon merging with a visualization system for repairing CAD models [90]. In addition, we provide a "bridge" [89] between CAD systems and layered manufacturing (rapid prototyping) systems. The system allows the user to accommodate CAD files, repair them, and consolidate multiple objects for one machine-running job.

We have also investigated a number of theoretical and applied issues regarding the general problem of storing and representing sets of polygons in three-dimensional space. We designed and implemented a software system [74] that supports storage of polyhedral data and of implementing algorithms on polyhedra. A three-dimensional polygon is a closed polygonal chain is 3D. We show [81] that determining its triangulability is NP-complete, and we show some sufficient and necessary conditions for triangulability.

In [31, 42, 41], we consider the practical problem of constructing binary space partitions (BSPs) for a set S of n orthogonal, non-intersecting, two-dimensional rectangles in three-dimensional Euclidean space such that the aspect ratio of each rectangle in S is at most α, for some constant $\alpha \geq 1$. We present an $O(n2^{\sqrt{\log n}})$-time algorithm to build a binary space partition of size $O(n2^{\sqrt{\log n}})$ for S. We also show that if m of the n rectangles in S have aspect ratios greater than α, we can construct a BSP of size $O(n\sqrt{m}2^{\sqrt{\log n}})$ for S in $O(n\sqrt{m}2^{\sqrt{\log n}})$ time. The constants of proportionality in the big-oh terms are linear in $\log \alpha$. We extend these results to cases in which the input contains non-orthogonal or intersecting objects. We also developed several approximation algorithms for surface, curve, and map simplification, a problem central to multiresolution modeling [54, 25, 114, 56]. We [38] developed a novel occlusion-culling algorithm that exploits the graphics hardware and the geometry of the input.

In [157], we explore a novel and practical method for back-face culling, which allows rendered scenes to be displayed faster in a visually-accurate manner. We continued this approach in [85] and [86], where we explore how to exploit coherence in interactive and fly-

through visual simulations of a polyhedral model to display silhouettes in highlight. This approach allows us to focus on the main factors that define the "shape" of an object in a simulation. Specifically, we present an efficient algorithm for computing a view-dependent silhouette of a polyhedral model. We compute perspective-accurate silhouettes by reducing the problem to point location queries, and demonstrate the usefulness of a fast silhouette-finder in rendering relatively large models, simplification and medical data registration.

In addition, we have developed a course [58] at SIGGRAPH that explores these topics of efficiency in the context of performing interactive walk-throughs in large geometric databases, particularly those of interest in defense settings.

3.5.5 Geometric Pattern Matching and Computational Metrology

Determining if a set of points (taken as samples from some manufactured part) conforms to a specification is essentially a geometric pattern matching problem. Defining and analyzing geometric models is often a first step in the manufacturing process, with the final step being that of matching that model to a set of sensor data (for either computational metrology or for automatic target recognition).

We initiated pioneering work that precisely defined the point-set pattern matching problem from an object-space point of view, and we gave several efficient deterministic algorithms for solving geometric pattern matching problems in this framework [111]. Moreover, we provide several efficient exact and approximation algorithms [125] for solving object-space geometric pattern matching problems that arise in computational metrology applications.

In addition, we define a new polygon-offset distance function and show how to efficiently construct Voronoi diagrams using this measure [83]. This distance-function is motivated by geometric pattern matching problems involving polygonal shapes, and we provide a number of applications [80] of our polygon-offset distance function definition and Voronoi diagrams defined by this distance to problems in computational metrology and robot motion planning. We produced a video [79] published in the video proceedings of the Symposium on Computational Geometry that was an animation of our algorithm for fitting a polygon to a set of points, as in a computational metrology application, by using a new "distance" measure that is defined by computing offsets from a reference polygon. We also describe two improved algorithms [84] for translating a convex polygon to contain a maximum number of points, which has applications in computer vision and geometric optimization, as well as provides a robust algorithm [91] for partial matching between surfaces and volumes, where the existence of no predefined features is assumed.

We present in [87] an efficient $O(n + 1/\epsilon^{4.5})$-time algorithm for computing a $(1+\epsilon)$-approximation of the minimum-volume bounding box of n points in $\Re^3$. We also present a simpler algorithm (for the same purpose) whose running time is $O(n \log n + n/\epsilon^3)$.

In [11, 77, 80] we give solutions to some constrained annulus placement problems for offset polygons and scaled polygons. Such matching problems show how closely a set of sample points matches a circular or polygonal boundary. The goal is to find the smallest annulus region of a circle or polygon containing a set of points subject to the following constraints: we fix the inner (resp., outer) polygon of the annulus and minimize the annulus region by minimizing the outer offset (resp., maximizing the inner offset). We also solve a problem that can be viewed as a special case of the first problem: finding the smallest translated

copy of a polygon containing an entire point set. For all of these problems, we solve for the cases where smallest and largest are defined by either offsetting the polygon or scaling it. In [14] we developed algorithms for computing the smallest cylinder enclosing a point set, and in [17] we developed an approximation algorithm for fitting a cylinder through a set of points.

The 3sum problem represents a class of problems conjectured to require $\Omega(n^2)$ time to solve, where n is the size of the input. Given two polygons P and Q in the plane, we show in [88] that some variants of the decision problem, whether there exists a transformation of P that makes it contained in Q, are 3sum-hard. We also show that the geometric pattern matching problem of finding the translation in the plane that minimizes the Hausdorff distance between two segment sets is 3sum-hard. Nevertheless, for two sets of points P and Q, we have designed several practical algorithms [146] for approximately matching P to a subset of Q under the Hausdorff distance subject to rotations, translations, and combined rotations and translations.

In [51] we developed efficient algorithms for exact pattern matching allowing translation and rotation. The analysis of our algorithm relies on a novel technique that bounds the number of simplices spanned by a set of points that are congruent to a given simplex. We also developed exact and approximation algorithms for minimum-weight matching, a fundamental problem in shape matching [173, 175].

In the same line of research, and in close collaboration with a foremost industry in the field (Marh/Federal of Providence, RI), we have developed an entirely new and eminently practical approach to a central problem in metrology, the determination of the *minimum zone cylinder* [119]. The minimum zone cylinder of a set of points in three dimensions is the cylindric crown defined by a pair of coaxial cylinders with minimal radial separation (width). In the context of tolerancing metrology, the set of points is nominally cylindrical, i.e., the points are known to lie in close proximity of a known reference cylinder. Using approximations which are valid only in the neighborhood of the reference cylinder, we can get a very good approximation of the minimum zone cylinder. The process provides successive approximations, and each iteration involves the solution of a linear programming problem in six dimensions. The error between the approximation and the optimal solution converges very rapidly (typically in three iterations in practice) down to a limit error of $\frac{8\omega_0^2}{R}$ (where ω_0 is the width and R is the external radius of the zone cylinder).

3.5.6 Layered Manufacturing

An important problem in *layered manufacturing* is the choice of a good build direction, which influences the quality and the cost of manufacturing the object. In [26], we present efficient algorithms for computing a build direction that optimizes the total area of faces that are in contact with supports.

4 Technology Transfer

Our research project has motivated the preliminary establishment of various collaborations worldwide. Also, attention has been already given by industry to the potential commercial

applications of our work. The collaborations will provide "real" users of our system prototypes and will leverage various research efforts funded by US and European agencies. The outreach of our Center for Geometric Computing is directed to both industries and U.S. Army (or Department of Defense) organizations.

With regard to industry, we have engaged in collaborative research with Marh/Federal in the area of computational metrology and with AT&T Research Laboratories on geometric techniques for storing and displaying large calling network graphs.

Our outreach to U.S. Army research projects has primarily been through the Army Research Lab in Aberdeen, although we have also been meeting and discussing research problems with people at ARL-Adelphi. We have also conducted discussions with scientists with other Department of Defense organizations, particularly DARPA and NSA. Specifically, our discussions have been directed at studying the feasibility of using our geometric data structures library for Internet security infrastructure tasks, such as certificate repository, as well as mechanisms for adding security to geometric collaboration tasks, such as exist in our GeomNet system.

Our efforts are also having an impact internationally. For example, our project on software development and Internet computing is likely to have an impact on the development of the CGAL library of geometric algorithms, a major European research project with seven sites. International collaborations with the INRIA French site (Jean-Daniel Boissonnat, director), the Saarbrücken German site (Kurt Mehlhorn, director), and the Israeli site (Micha Sharir, director) have been activated and materialized in productive visits.